**Python:eval的妙用和滥用**

2014年07月08日 13:05:24

阅读数：41940

eval()函数十分强大，官方demo解释为：将字符串str当成有效的表达式来求值并返回计算结果。

so，结合math当成一个计算器很好用。

其他用法，可以把list,tuple,dict和string相互转化。见下例子：

**[python]** [view plain](https://blog.csdn.net/zhanh1218/article/details/37562167) [copy](https://blog.csdn.net/zhanh1218/article/details/37562167)

1. a = "[[1,2], [3,4], [5,6], [7,8], [9,0]]"
3. b = eval(a)
5. b
6. Out[3]: [[1, 2], [3, 4], [5, 6], [7, 8], [9, 0]]
8. type(b)
9. Out[4]: list
11. a = "{1: 'a', 2: 'b'}"
13. b = eval(a)
15. b
16. Out[7]: {1: 'a', 2: 'b'}
18. type(b)
19. Out[8]: dict
21. a = "([1,2], [3,4], [5,6], [7,8], (9,0))"
23. b = eval(a)
25. b
26. Out[11]: ([1, 2], [3, 4], [5, 6], [7, 8], (9, 0))

不可谓不强大！

ＢＵＴ！强大的函数有代价。安全性是其最大的缺点。

想一想这种使用环境：需要用户输入一个表达式，并求值。

如果用户恶意输入，例如：

**[plain]** [view plain](https://blog.csdn.net/zhanh1218/article/details/37562167) [copy](https://blog.csdn.net/zhanh1218/article/details/37562167)

1. \_\_import\_\_('os').system('dir')

那么eval()之后，你会发现，当前目录文件都会展现在用户前面。

那么继续输入：

**[plain]** [view plain](https://blog.csdn.net/zhanh1218/article/details/37562167) [copy](https://blog.csdn.net/zhanh1218/article/details/37562167)

1. open('文件名').read()

代码都给人看了。获取完毕，一条删除命令，文件消失。哭吧！

怎么避免安全问题？

１、自行写检查函数；

２、使用ast.literal\_eval：[自行查看DOCUMENT](https://docs.python.org/2/library/ast.html" \t "_blank)

3、更多好文：[Restricted "safe" eval(Python recipe)](http://code.activestate.com/recipes/496746-restricted-safe-/" \t "_blank)

本文由@The\_Third\_Wave（Blog地址：<http://blog.csdn.net/zhanh1218>）原创。还有未涉及的，会不定期更新，有错误请指正。

如果你看到这篇博文时发现不完整，那是我为防止爬虫先发布一半的原因，请看原作者Blog。

如果这篇博文对您有帮助，为了好的网络环境，不建议转载，建议收藏！如果您一定要转载，请带上后缀和本文地址。